

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method, comprising:
defining a plurality of first rules made up of relatively less complex second rules;
defining a policy having at least some of the plurality of first rules and defining corresponding actions to undertake that are related to communication of a packet within a network;
converting the first rules into minterm representations;
generating a bit mask for each of the second rules based on their presence in the minterm representations;
using the generated bit masks and content in a header of the packet to evaluate the plurality of first rules in the policy and to determine a corresponding action to undertake.
2. (Original) The method of claim 1 wherein using the content in the header of the packet to evaluate the plurality of first rules includes using content in fields of a hypertext transfer protocol (HTTP) header of the packet.
3. (Original) The method of claim 2 wherein using content in the fields of the HTTP header of the packet includes using content from at least one of header value, header name, universal resource locator string, method, hostname, cookie, defined, and undefined fields of the HTTP header.
4. (Original) The method of claim 1 wherein defining the plurality of first rules made up of relatively less complex second rules includes defining the first rules by nesting the second rules using logical operators.

5. (Original) The method of claim 1 wherein defining the corresponding actions to undertake that are related to communication of the packet within the network includes at least one of defining forward, redirect, persist, reply error, and reset client actions for each set of matching first rules in the policy.

6. (Currently Amended) The method of claim 1 wherein converting the first rules into minterm representations includes converting the first rules into minterms ~~having sums of that are~~ products of the second rules, said products being summed together, and storing the minterms in a minterm data structure.

7. (Currently Amended) The method of claim 6, further comprising placing all second rules of a similar type in same rule type data structures, wherein generating the bit mask for each of the second rules based on their presence in the minterm representations includes:

determining a total number of minterms in the policy and using the determined total number of minterms to define a number of bit positions for the bit mask; and

for each of the second rules:

determining whether a particular second rule is present in each successive minterm in the minterm data structure ~~for each of the rule type databases~~, and enabling a bit position of the bit mask that corresponds to each minterm ~~where in which~~ that particular second rule is present;

disabling a bit position of the bit mask that corresponds to each minterm ~~where in which~~ that particular second rule is not present; and

enabling a bit position of the bit mask that corresponds to each minterm ~~where in which~~ that particular second rule is not present ~~and where the minterm~~ but includes second rules from a different rule type ~~database~~ data structure.

8. (Currently Amended) The method of claim 7 wherein using the generated bit masks and content in the header of the packet to evaluate the plurality of first rules in the policy and to determine the corresponding action to undertake includes:

searching for a second rule in each rule type ~~database~~ data structure that corresponds to content in the packet;

for second rules located by the searching, obtaining the corresponding generated bit masks;

applying a logical operation to the obtained bit masks to generate a new bit mask;

locating a first non-zero value in a bit position of the new bit mask, and designating a first rule ~~in-associated with~~ a minterm corresponding to that bit position as a match for said corresponding action to undertake.

9. (Original) The method of claim 8 wherein the logical operation includes an AND operation.

10. (Currently Amended) A method, comprising:
~~defining-storing~~ a plurality of complex rules made up of simpler rules and being indicative of actions to take relative to processing of a packet communicated within a network;

~~converting-storing minterm representations obtained from~~ the complex rules ~~into minterm representations~~;

~~generating-storing~~ a bit mask generated for each simpler rule based on the minterm representations of the complex rules;

examining header content of the packet; and

using the header content of the packet and the bit masks to evaluate the complex rules represented as minterms, and determining which action to undertake relative to that packet in accordance with results of the evaluation.

11. (Currently Amended) The method of claim 10 wherein using the header content of the packet and the bit masks to evaluate the complex rules represented as minterms includes:

determining which simpler rules correspond to the header content;
determining the bit masks for these determined simpler rules;
performing a logical operation on these determined bit masks to generate a new bit mask; and

determining a complex rule that matches the header content from a minterm identifiable from the new bit mask.

12. (Currently Amended) The method of claim 10 wherein ~~generating~~ the bit mask generated for each simpler rule includes ~~generating~~ bit position values in the bit mask based on whether a particular simpler rule is present in a minterm.

13. (Currently Amended) The method of claim 10, further comprising:
~~placing~~ storing simpler rules of similar rule type in a common data structure; and
searching each data structure for a specific simpler rule that corresponds to the header content.

14. (Currently Amended) The method of claim 10 wherein ~~defining~~ the plurality of complex rules made up of simpler rules includes ~~using~~ a plurality of logical operators to relate a plurality of simpler rules to form at least one complex rule.

15. (Currently Amended) A method, comprising:
~~reducing~~ storing a first rule into at least one minterm obtained from a first rule
and made of a plurality of second rules that are less complex relative to the first rule;
~~generating~~ storing a bit mask for each of the second rules;
for data from any field in a header of a packet, determining which second rules correspond to that data;

applying a logical operation to stored the bit masks of the second rules corresponding to the data to obtain a new bit mask; and

determining an action to undertake related to the packet from a minterm validated via the new bit mask.

16. (Original) The method of claim 15 wherein applying the logical operation to the bit masks of the second rules to obtain the new bit mask includes applying an AND operation to these bit masks, and wherein determining the action to undertake from the minterm validated via the new bit mask includes identifying the minterm from a first non-zero bit position in the new bit mask.

17. (Currently Amended) The method of claim 15 wherein ~~generating~~ the bit mask for each of the second rules includes ~~determining~~ bit position values of the bit mask based on whether a particular second rule is present in a given minterm.

18. (Original) The method of claim 15, further comprising:
examining data in fields of the header in addition to hostname, URL, and cookie fields; and

searching for second rules corresponding to this data in separate data structures organized according to rule types.

19. (Currently Amended) An article of manufacture, comprising:
a machine-readable storage medium having instructions stored thereon and
executable by a processor to:

define a plurality of first rules made up of relatively less complex second rules;
define a policy having at least some of the plurality of first rules and define corresponding actions to undertake that are related to communication of a packet within a network;

convert the first rules into minterm representations;

generate a bit mask for each of the second rules based on their presence in the minterm representations;

use the generated bit masks and content in a header of the packet to evaluate the plurality of first rules in the policy and to determine a corresponding action to undertake.

20. (Currently Amended) The article of manufacture of claim 19 wherein the instructions to convert the first rules into minterm representations includes instructions to convert the first rules into minterms ~~having sums of~~ that are products of the second rules, said products being summed together, and to store the minterms in a minterm data structure,

wherein the machine-readable medium includes instructions stored thereon to locate all second rules of a similar type in same rule type data structures,

wherein the instructions to generate the bit mask for each of the second rules based on their presence in the minterm representations includes instructions to:

determine a total number of minterms in the policy and use the determined total number of minterms to define a number of bit positions for the bit mask; and

for each of the second rules:

determine whether a particular second rule is present in each successive minterm in the minterm data structure ~~for each of the rule type databases~~, and enable a bit position of the bit mask that corresponds to each minterm ~~where~~ in which that particular second rule is present;

disable a bit position of the bit mask that corresponds to each minterm ~~where~~ in which that particular second rule is not present; and

enable in a bit position of the bit mask that corresponds to each minterm ~~where~~ in which that particular second rule is not present ~~and where the minterm~~ but includes second rules from a different rule type ~~database~~ data structure.

21. (Currently Amended) The article of manufacture of claim ~~49-20~~ wherein the instructions to use the generated bit masks and content in the header of the packet to evaluate

the plurality of first rules in the policy and to determine the corresponding action to undertake includes instructions to:

search for a second rule in each rule type ~~database~~ data structure that corresponds to content in the packet;

for second rules located by the searching, obtain the corresponding generated bit masks;

apply a logical operation to the obtained bit masks to generate a new bit mask;

locate a first enabled bit position of the new bit mask, and designate a first rule ~~in~~ associated with a minterm corresponding to that bit position as a match for said action to undertake.

22. (Original) A system, comprising:

a means for defining a plurality of complex rules made up of simpler rules and being indicative of actions to take relative to processing of a packet communicated within a network;

a means for converting the complex rules into minterm representations;

a means for generating a bit mask for each simpler rule based on the minterm representations of the complex rules;

a means for examining header content of the packet; and

a means for using the header content of the packet and the bit masks to evaluate the complex rules represented as minterms, and for determining which action to undertake relative to that packet in accordance with results of the evaluation.

23. (Original) The system of claim 22 wherein the means for using the header content of the packet and the bit masks to evaluate the complex rules represented as minterms includes:

a means for determining which simpler rules correspond to the header content;

a means for determining the bit masks for these simpler rules;

a means for performing a logical operation on these bit masks to generate a new bit mask; and

a means for determining a complex rule that matches the header content from a minterm identifiable from the new bit mask.

24. (Original) The system of claim 22 wherein the means for generating the bit mask for each simpler rule includes a means for generating bit position values in the bit mask based on whether a particular simpler rule is present in a minterm, the apparatus further comprising:

a means for placing simpler rules of similar rule type in a common data structure;
and

a means for searching each data structure for a specific rule that corresponds to the header content.

25. (Original) The system of claim 22, further comprising a means for communicating between a client device and a network component.

26. (Original) An apparatus, comprising:

a data structure having a first rule reduced into at least one minterm made of a plurality of second rules, and having a bit mask generated for each of the second rules;

a first component having access to the data structure to determine which second rules correspond to data from any field in a header of a packet;

a second component to apply a logical operation to the bit masks of the second rules determined by the first component to correspond to the data, and to obtain a new bit mask as a result of application of the logical operation; and

a third component having access to the data structure to determine an action to undertake related to the packet from a minterm validated via the new bit mask.

27. (Original) The apparatus of claim 26 wherein at least one of the data structure, first component, second component, and third component are located in a switch that can receive the packet.

28. (Original) The apparatus of claim 26 wherein the data structure includes a plurality of rule type data structures that correspondingly store second rules of similar rule types.

29. (Original) The apparatus of claim 26 wherein the header comprises an HTTP header of the packet.

30. (Original) The apparatus of claim 26 wherein to define the first rule, the second rules are related using a plurality of logical operators.

31. (New) The method of claim 10 wherein said minterm representations are stored in a minterm data structure, the method further comprising:

storing all simpler rules of a similar type in same rule type data structures,
wherein the bit mask generated for each simpler rule includes:

a) a number of bit positions determined according to a total number of minterm representations in said minterm data structure; and

b) enabled and disabled bit positions based on: whether the simpler rule corresponding to the bit mask is present in a minterm representation respectively associated with each bit position, and whether the simpler rule corresponding to the bit mask is not present in the minterm representation respectively associated with each bit position and other simpler rules present in said minterm representation are stored in a different rule type data structure.

32. (New) The method of claim 15 wherein storing said at least one minterm includes storing a plurality of minterms in a minterm data structure, the method further comprising:

storing all second rules of a similar type in same rule type data structures,
wherein the bit mask for each second rule includes:

a) a number of bit positions determined according to a total number of minterms in said minterm data structure; and

b) enabled and disabled bit positions based on: whether the second rule corresponding to the bit mask is present in a minterm respectively associated with each bit position, and whether the second rule corresponding to the bit mask is not present in the minterm respectively associated with each bit position and other second rules present in said minterm are present in a different rule type data structure.

33. (New) The system of claim 22, further comprising:

minterm data structure means for storing said minterm representations; and

multiple rule type data structure means for respectively storing all simpler rules of a similar rule type,

wherein said means for generating the bit mask for each simpler rule includes:

a) means for determining a total number of minterm representations in said minterm data structure means and for using the determined total number of minterm representations to define a number of bit positions for the bit mask; and

b) for each of the simpler rules:

means for determining whether a particular simpler rule is present in each successive minterm representation in the minterm data structure means, and for enabling a bit position of the bit mask that corresponds to each minterm representation in which that particular simpler rule is present;

means for disabling a bit position of the bit mask that corresponds to each minterm representation in which that particular simpler rule is not present; and

means for enabling in a bit position of the bit mask that corresponds to each minterm representation in which that particular simpler rule is not present but includes at least one simpler rule of a different rule type.

34. (New) The apparatus of claim 26 wherein said data structure is adapted to

separately store together all second rules of a similar rule type, and

wherein the bit mask for each of the second rules includes:

a) a number of bit positions determined according to a total number of minterms in said data structure; and

b) enabled and disabled bit positions based on: whether the second rule corresponding to the bit mask is present in a minterm respectively associated with each bit position, and whether the second rule corresponding to the bit mask is not present in the minterm respectively associated with each bit position and other second rules of a different rule type are present in said minterm.

35. (New) An apparatus, comprising:

at least one storage medium configured to store:

a minterm table having a plurality of minterms each associated with one of a plurality of first rules that are each defined according to a logical relationship of second rules, each minterm including products of said second rules said first rules, each of said first rules specifying a routing action to apply to a packet;

a plurality of bit masks respectively corresponding to each of said second rules; and

a processor coupled to said at least one storage medium and adapted to: identify matching ones of said second rules that match header content of said packet, to perform a logical operation between bit masks corresponding to said matching second rules to obtain a new bit mask, to use an enabled bit position in said new bit mask to identify from said minterm table a valid one of said first rules, and to apply the routing action specified by said identified valid first rule to said packet.

36. (New) The apparatus of claim 35 wherein said at least one storage medium is further configured to store different types of said second rules in separate data structures so that similar types of said second rules are stored in same data structures.

37. (New) The apparatus of claim 35 wherein each of said bit masks has a number of bit positions corresponding to a number of minterms in said minterm table, each bit position being associated with a respective minterm in said minterm table, wherein a value of each bit position of each bit mask is based at least in part on whether the second rule corresponding to the bit mask is present in the respective minterm associated with each bit position.

38. (New) The apparatus of claim 37 wherein a value of a bit position is binary 1 if said corresponding second rule is present in the respective minterm associated with that bit position, binary 0 if said corresponding second rule is not present in the respective minterm associated with that bit position, and is binary 1 if said corresponding second rule is not present in the respective minterm associated with that bit position and if said respective minterm associated with that bit position includes second rules that are of a different type than said corresponding second rule.

39. (New) The apparatus of claim 35 wherein said enabled bit position, in said new bit mask used by said processor to identify said valid first rule, is first bit position in said new bit mask having a non-zero value, said bit position having said non-zero value being usable by said processor as an index to said minterm table to identify said valid first rule.

40. (New) The apparatus of claim 35 wherein to perform said logical operation between said bit masks corresponding to said matching second rules, said processor is adapted to perform a logical AND operation between values of corresponding bit positions in said bit masks.